10

Liberando el Acceso login #1

Transcripción

[00:00] Ya tenemos nuestro flujo de autorización completo, pero hay algunas cosas que aún necesitamos implementar aquí para que no solo sea completo, sino que sea óptimo y esté bien hecho, porque hay algunos casos de uso que se nos están escapando y los vamos a ver ahora.

[00:15] Por ejemplo, ella siendo un pequeño recap ya hemos visto cómo obtener el subject, estamos validando la firma del token, que estamos recibiendo, el header. Estamos validando el issuer, verificamos el token en sí y al final si el token es inválido retornamos alguna excepción y ahí terminaría todo nuestro proceso hasta ahora.

[00:37] Entonces si venimos a nuestra recolección de request, aquí, por ejemplo, yo le voy a dar send. Vemos que estoy recibiendo la información solicitada a mis recursos porque yo estoy enviando mi token de autorización aquí en header. Entonces funciona bien mi caso de uso en este caso.

[00:58] Si quiero por ejemplo obtener datos médicos yo no tengo ningún tipo de autenticación configurado, por lo tanto me debería dar error, vamos a ver aquí y en efecto miren aquí, me dio 500 internal server error, porque lo que él hace es un throw new runtime exception, y miren, el token enviado no es válido.

[01:17] Entonces está validando que mi token no está siendo enviado y eso es bueno, eso es lo que yo quería, ese era el objetivo principal, pero ahora miren aquí a login. En login estamos obteniendo el token aquí, pero si yo quiero generar ahora otro token, miren el error que me da.

[01:37] Ahora también me dice que el token no es válido y aquí sí estamos en un grave error, porque login tiene que ser abierto para todo mundo, de lo contrario es completamente imposible generar autorization tokens. Entonces eso es lo primero que vamos a arreglar ahora en esta parte. Para eso vamos a nuestro SecurityConfigurations.

[01:57] Y aquí en nuestro security FilterChain. Ya sabemos que aquí nosotros le decimos a Spring, le indicamos Springs el tipo de sesión. Y ahora lo que necesitamos es decirle que queremos autorizar los requests que tengan el siguiente matching por ejemplo los requests que lleguen con HttpMethod, get no, yo quiero un post.

[02:33] Y los requests que comiencen con login. Este va hacer el pattern, el patrón que va a seguir. Con eso le estoy diciendo que los requests hagan match con que tiene que ser un post y que sea login. Y también lo que yo quiero es que aquí le permita todo, entonces le doy un .permitAll().

[02:56] Entonces ahí con esto yo ya debería haber abierto mi login para todo mundo porque yo estoy diciendo que si también en mi request matchers, si viene un método que es un post y va a login, entonces permitirle a todos. De lo contrario y aquí viene otro, anyRequest .authenticated.

[03:21] Entonces, ¿cómo se lee esto? Esto se lee como decirle Spring, la política de creación es stateless y cada request que haga match que es un request de tipo post y va para login permitirle a todos. Después todos los requests tienen que ser autenticados. Y bueno, construye el objeto finalmente.

[03:46] Entonces esta es la forma, como le podemos decir a Spring: “abre este recurso para que sea público para todo mundo”. Vamos a probar si está funcionando. Entonces limpiamos, esperamos un momento, vemos que ya recargó. Listo. Y vamos a intentar otra vez. Le vamos a dar send y me va a dar un 403 forbidden, vemos que ya no me da la excepción, pero ahora me está diciendo que no tengo acceso a este recurso.

[04:18] Entonces vamos a investigar por qué. ¿Qué es lo que me dice aquí el error de Spring? Lo que él me dice el token enviado no es válido, o sea, estamos aún cayendo en el error de validación del token de Spring, pero ahora tú me preguntas: “Diego, pero se supone que estamos permitiendo eso”.

[04:36] El problema está exactamente en cómo estamos manejando eso con los filtros, porque en nuestro filter, si vemos aquí entonces si no recibe esto simplemente no lo toma como válido. Tenemos que indicarle aquí que el usuario que ha enviado ese token en realidad es válido. Para login debería ser abierto, simplemente abierto para todos los requests que quiera hacer. Entonces vamos a ver ese problema ahora.